# ProbLog Technology for Inference in a Probabilistic First Order Logic

Luc De Raedt
Katholieke Universiteit Leuven (Belgium)

joint work with Maurice Bruynooghe, Theofrastos Mantadelis, Angelika Kimmig , Bernd Gutmann, Gerda Janssens, and Joost Vennekens

ECAI 2010

# Motivation

- Probabilistic logic programming formalisms such as PHA & ICL (Poole), PRISM (Sato), ProbLog (De Raedt et al.)
  - extend Prolog with probabilistic facts, clauses are deterministic (hard)
  - restricted to Prolog, no full first order theories
  - inference based on SLD-resolution (theorem proving)
- Markov Logic (Domingos et al.)
  - extends Markov networks with first order logic, clauses are soft constraints
  - inference based on maxSAT a.o.
- Can we combine these ideas ?
  - soft constraints / probabilistic clauses
  - theorem proving

# Outline

1. ProbLog
2. FOProbLog
3. From FOProbLog to ProbLog
4. Case studies
5. Conclusion

# ProbLog

## An example

$likes(X) :- red(X).$
$likes(X) :- round(X).$
$0.8 :: round(X).$
$0.3 :: red(X).$

# ProbLog

## An example

$likes(X) :- red(X).$
$likes(X) :- round(X).$
$0.8 :: round(X).$
$0.3 :: red(X).$

## Four possible worlds for constant $a$

$0.8 \times 0.3 : \{round(a), red(a), likes(a)\}$
$0.2 \times 0.3 : \{red(a), likes(a)\}$
$0.8 \times 0.7 : \{round(a), likes(a)\}$
$0.2 \times 0.7 : \{\}$
Total choice / Belief sets

# ProbLog

## An example

$likes(X) :- red(X).$
$likes(X) :- round(X).$
$0.8 :: round(X).$
$0.3 :: red(X).$

## Computation

$$
\begin{aligned}
\text{Prob } likes(a)) &= \text{Prob}(round(a) \vee red(a)) \\
&= \text{Prob } round(a) + \text{Prob}(\neg round(a) \wedge red(a)) \\
&= 0.8 + (1 - 0.8) \times 0.3
\end{aligned}
$$

# ProbLog Technology

- Collect proofs
- Proofs not necessarily disjoint
- create BDD to cope with disjoint sum problem
- compute/approximate probability
- ProbLog integrated in YAP Prolog, download from http://dtai.cs.kuleuven.be/problog/

# ProbLog Concepts

- Facts with probabilities
- Belief set: a subset of the facts
  Has a probability
- Semantics: Least Herbrand model
- Inference: probability of a ground atom in a randomly selected belief set

# Outline

# From ProbLog to FOProbLog

## What If?

FO formulas instead of definite clauses?

## Problems

- SLD proof procedure is not complete
- belief set can be inconsistent

# Example

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

# Example

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

# Example

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

$\forall x : male(x) \wedge female(x) \rightarrow false$
no choice, probability is 1

# Example cont.

## A belief set

$female(Floris) : 0.6$

$\forall x : cs(x) \rightarrow male(x) : 0.8$

$\forall x : male(x) \wedge female(x) \rightarrow false$

probability 0.6*0.8=0.48

We can infer $\neg cs(Floris)$

This is not ProbLog

# Example cont.

$$\mathit{male}(\mathit{Floris}) : 0.4 \quad \vee \quad \mathit{female}(\mathit{Floris}) : 0.6$$
$$\forall x : \mathit{cs}(x) \rightarrow \mathit{male}(x) : 0.8 \quad \vee \quad \mathit{cs}(x) \rightarrow \mathit{female}(x) : 0.2$$
$$\forall x : \mathit{male}(x) \wedge \mathit{female}(x) \rightarrow \mathit{false}$$
$$\mathit{cs}(\mathit{Floris})$$

# Example cont.

$$
\begin{aligned}
&male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6\\
&\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2\\
&\forall x : male(x) \wedge female(x) \rightarrow false\\
&cs(Floris)
\end{aligned}
$$

Inconsistent belief set with probability 0.48
Compute probability of inconsistent belief sets
Redistribute probability mass over consistent belief sets.

# Outline

1. ProbLog
2. FOProbLog
3. From FOProbLog to ProbLog
4. Case studies
5. Conclusion

# How to implement?

## How to do inference?

- # belief sets is exponential in # of choices between ground formulas
- No way to enumerate them all

## Which technology?

- Can we preserve ProbLog Technology?
- How to collect the proofs?
- Can we preserve Prolog technology for that?

# How to implement?

## How to do inference?

- $\#$ belief sets is exponential in $\#$ of choices between ground formulas
- No way to enumerate them all

## Which technology?

- Can we preserve ProbLog Technology?
- How to collect the proofs?
- Can we preserve Prolog technology for that?
- Yes we can

# How to implement?

## How to do inference?

- $\#$ belief sets is exponential in $\#$ of choices between ground formulas
- No way to enumerate them all

## Which technology?

- Can we preserve ProbLog Technology?
- How to collect the proofs?
- Can we preserve Prolog technology for that?
- Yes we can
- Stickel's Prolog Technology Theorem Prover

# Translation to ProbLog

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

# Translation to ProbLog

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

```
0.4::pf_fl(floris).% probabilistic fact
```

# Translation to ProbLog

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

```
0.4::pf_fl(floris).% probabilistic fact
male(floris):-pf_fl(floris).
female(floris):-not(pf_fl(floris)).% negation
```

# Translation to ProbLog

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

# Translation to ProbLog

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

```
0.8::pf_cs(X). % probabilistic fact
```

# Translation to ProbLog

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

```
0.8::pf_cs(X). % probabilistic fact
male(X):-cs(X), pf_cs(X).
not_cs(X):-not_male(X), pf_cs(X). % contrapositive
```

# Translation to ProbLog

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

```
0.8::pf_cs(X). % probabilistic fact
male(X):-cs(X), pf_cs(X).
not_cs(X):-not_male(X), pf_cs(X). % contrapositive
female(X):-cs(X), not(pf_cs(X)).
not_cs(X):-not_female(X), not(pf_cs(X)).
```

# Translation to ProbLog

$\forall x : male(x) \wedge female(x) \rightarrow false$
$cs(Floris)$

# Translation to ProbLog

$\forall x : male(x) \wedge female(x) \rightarrow false$

$cs(Floris)$

```
not_female(X):-male(X).
not_male(X):-female(X).
```

# Translation to ProbLog

$\forall x : male(x) \wedge female(x) \rightarrow false$

$cs(Floris)$

```
not_female(X):-male(X).
not_male(X):-female(X).
cs(floris).
```

# SLD is incomplete and depth first

## Stickel: ancestor resolution makes it complete

While proving inconsistency for `p(t)`
A subgoal `not_p(t)` is inconsistent with `p(t)`
Hence can be dropped.

And similar for `not_p(t)` and `p(t)`

## Stickel: iterative deepening avoids infinite branches

## Solution

Modify the SLD engine
Not so different from tabling
Complicates tabling!

# Some Formulas

Total choice: Making a decision for every probabilistic fact
Corresponds to selection of a belief set

**normalized probability of a total choice $\widehat{prob}(s)$**

$s$: a total choice
$Cons$: total choices that result in consistent belief set
$InCons$: total choices that result in inconsistent belief set

$$\widehat{prob}(s) = prob(s) / \sum_{s \in Cons} prob(s) \text{ for } s \in Cons$$
$$= prob(s)/1 - \sum_{s \in InCons} prob(s)$$
$$\widehat{prob}(s) = 0 \text{ otherwise}$$

Constraint on probability distribution

# Minimal probability of a query

## Probability distribution is not unique

$pf(a) : 0.7.$
$p(a) : -pf(a).$
The empty total choice (`pf(a)` is false) has probability 0.3
Allows for two models: $\emptyset$ and $\{p(a)\}$

Hence the probability of a query $Q$ has a minimum and a maximum.
Maximum probability of Q is minimum probability of $\neg Q$

## Theorem: Minimal probability of a query

$\min_{\mu \in \hat{\mathcal{M}}} \mu(Q) = \sum_{s \models Q} \widehat{prob}(s)$
where $s \models Q$ means that $Q$ can be proven in $s$

# Proving inconsistency by running `?- false.`

## A naive way

```
false:-male(X), not_male(X).
false:-female(X), not_female(X).
false:-cs(X), not_cs(X).
```

A lot of redundant proofs

## Starting from negative clauses

```
false:-male(X), female(X).
```

## Starting from positive clauses

```
false:-not_male(floris), pf_fl(floris).
false:-not_female(floris), not(pf_fl(floris)).
false:-not_cs(floris).
```

# Answering queries

- Proofs are partial choices
- The formula/BDD $\phi$ represents all total choices that extend those partial choices
- This includes inconsistent total choices
- With $\psi$ the formula/BDD for the query *false*, $\phi \wedge \neg\psi$ corresponds to the proofs made up from consistent total choices.

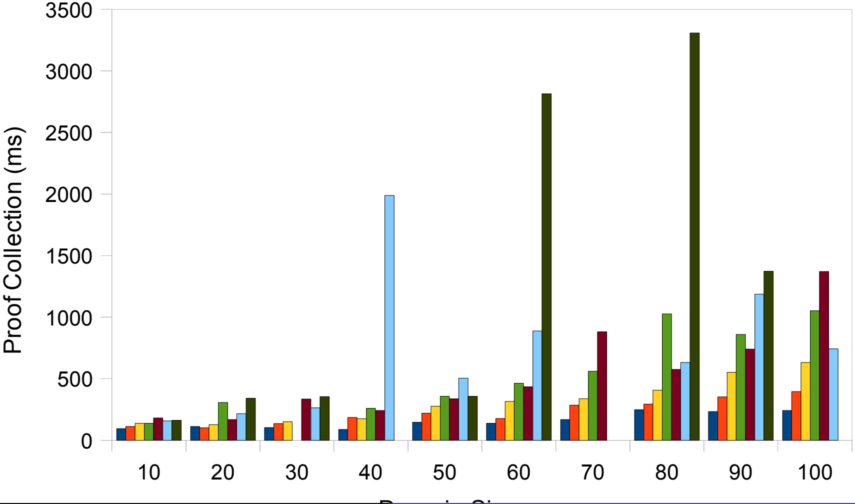
# Outline

1. ProbLog
2. FOProbLog
3. From FOProbLog to ProbLog
4. Case studies
5. Conclusion

# Friends of friends

$\forall x, y, z, pf1(x, y, z) \rightarrow (Fr(x, y) \wedge Fr(y, z) \rightarrow Fr(y, z))$
$\forall x, pf2(x) \rightarrow (Smokes(x) \rightarrow Cancer(x))$
$\forall x, y, pf3(x, y) \rightarrow (Fr(x, y) \rightarrow (Smokes(x) \leftrightarrow Smokes(y)))$
Experiments with growing domain size and depth bound

# Friends of friends



# Entity resolution

## An MLN application

Parag Singla and Pedro Domingos, 'Entity resolution with Markov Logic', in *ICDM 2006*, pp. 572–582.

### A database

$author(paper, author)$
$title(paper, title)$
$venue(paper, venue)$

# Entity resolution

## An MLN application

Parag Singla and Pedro Domingos, ‘Entity resolution with Markov Logic’, in *ICDM 2006*, pp. 572–582.

### A database

$author(paper, author)$
$title(paper, title)$
$venue(paper, venue)$
$hasWordAuthor(author, word)$
$hasWordTitle(title, word)$
$hasWordVenue(venue, word)$

1295 bibliographic entries involving roughly 90 authors, 400 venues, 200 titles and 2700 words

# Closed World Assumption on the Database

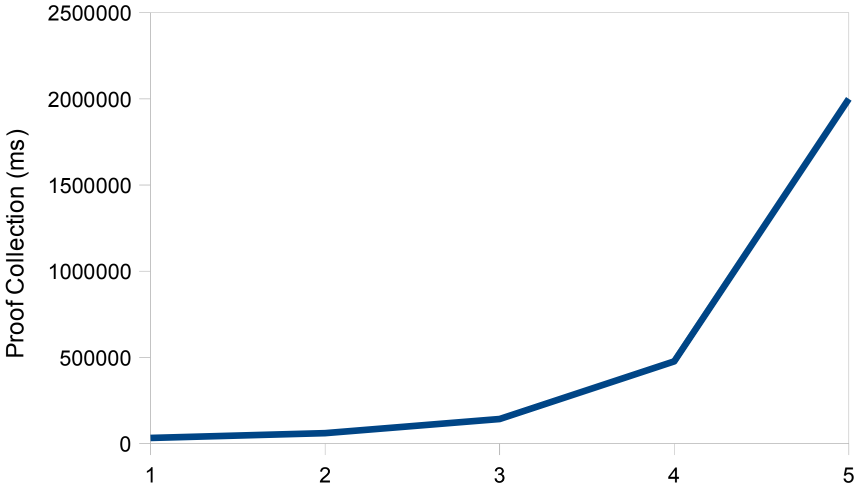
negation as finite failure
(calls have to be ground!)

```
not_author(B,A):-not(author(B,A)).
not_title(B,T):-not(title(B,T)).
not_venue(B,V):-not(venue(B,V)).
not_hasWordAuthor(A,W):-not(hasWordAuthor(A,W)).
not_hasWordTitle(T,W):-not(hasWordTitle(T,W)).
not_hasWordVenue(V,W):-not(hasWordVenue(V,W)).
```

# Clauses like

- Publications can be the same because they share authors
- Authors can be the same because they share publications
- Authors can be the same because their names share words
- Titles can be the same because their names share words
- Venues can be the same because their names share words

# Entity Resolution



# Outline

# Conclusion

- Elegant formalism. Real probabilities.
- Can express Nilssons's logic: $F : p \vee \neg F : 1 - p$
- Theorem proving for probabilistic logic
- # proofs typically exponential in depth of search
- Entitity resolution application beyond current ProbLog implementation (normalisation requires to run `?-false.`)
- Avoid redundancy and inconsistency in theory
- Would be interesting to develop a sampling approach